EasyAVR® 6

Manual de usuario

Sistema de desarrollo

Todos los sistemas de desarrollo de MikroElektronika son unas herramientas insustituibles para la programación y el desarrollo de los dispositivos basados en microcontroladores. Las componentes elegidas con atención debida y el uso de las máquinas de la última generación para montarlas y probarlas son la mejor garantía de alta fiabilidad de nuestros dispositivos. Gracias a un diseño simple, un gran número de los módulos complementarios y los ejemplos listos para ser utilizados todos nuestros usuarios, sin reparar en su experiencia, tienen la posibilidad de desarrollar sus proyectos en una manera fácil y eficiente.

mikroElektronika
SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

ESTIMADOS CLIENTES,

Querría darles las gracias por estar interesados en nuestros productos y por tener confianza en MikroElektronika.

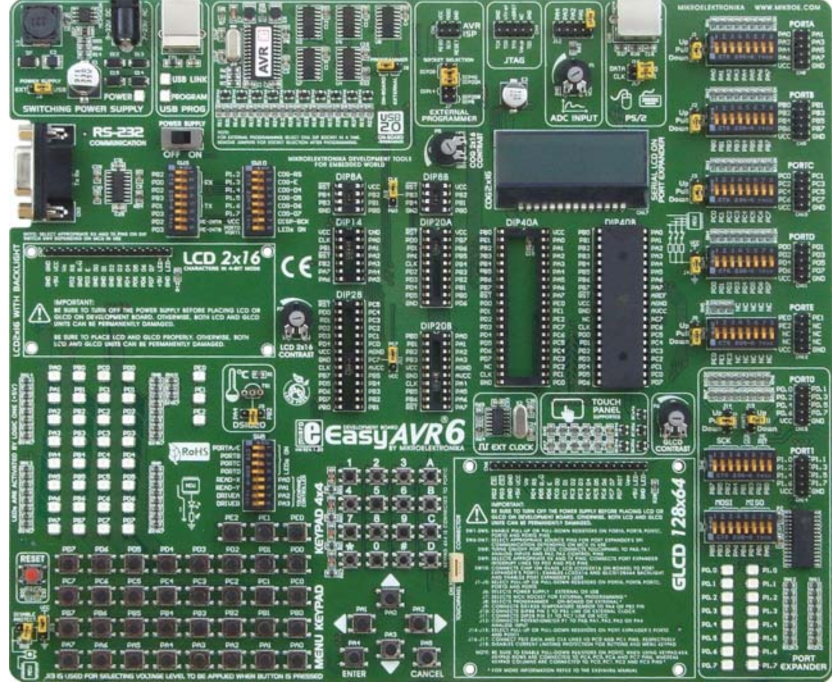
Nuestro objetivo es proporcionarles con los productos de la mejor calidad. Además, seguimos mejorando nuestros rendimientos para responder a sus necesidades.

Nebojsa Matic
Director general

TABLA DE CONTENIDO

Introducción a la placa de desarrollo EasyAVR6

El sistema de desarrollo *EasyAVR6* es una herramienta de desarrollo extraordinaria, adecuada para la programación y la experi-men- tación con los microcontroladores AVR® de la compañía *Atmel*®. Este sistema dispone de un programador incorporado que proporciona una interfaz entre el microcontrolador y el PC. Se espera de Ud. que escriba un código en alguno de nuestros compila-dores, que genere un fichero .hex y que programe el microcontrolador utilizando el programador *AVRprog*®. Los numerosos módulos incorporados como visualizador gráfico LCD de 128x64 píxeles, visualizador alfanumérico LCD de 2x16 caracteres, visualizador LCD integrado en la placa de 2x16 caracteres con comunicación serial, teclado 4x4, expansor de puertos etc, le permiten simular con facilidad el funcionamiento del dispositivo destino.





Placa de desarrollo completa y fácil de utilizar para los microcontroladores AVR.



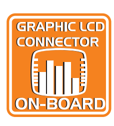
Programador integrado en la placa de altas prestaciones con comunicación USB 2.0.



Expansor de puertos incorporado proporciona expansión de E/S (2 puertos adicionales) utilizando conversión de formato de datos.



Visualizador alfanumerico LCD integrado en la placa de 2x16 caracteres con comunicación serial



Visualizador gráfico LCD con iluminación de fondo



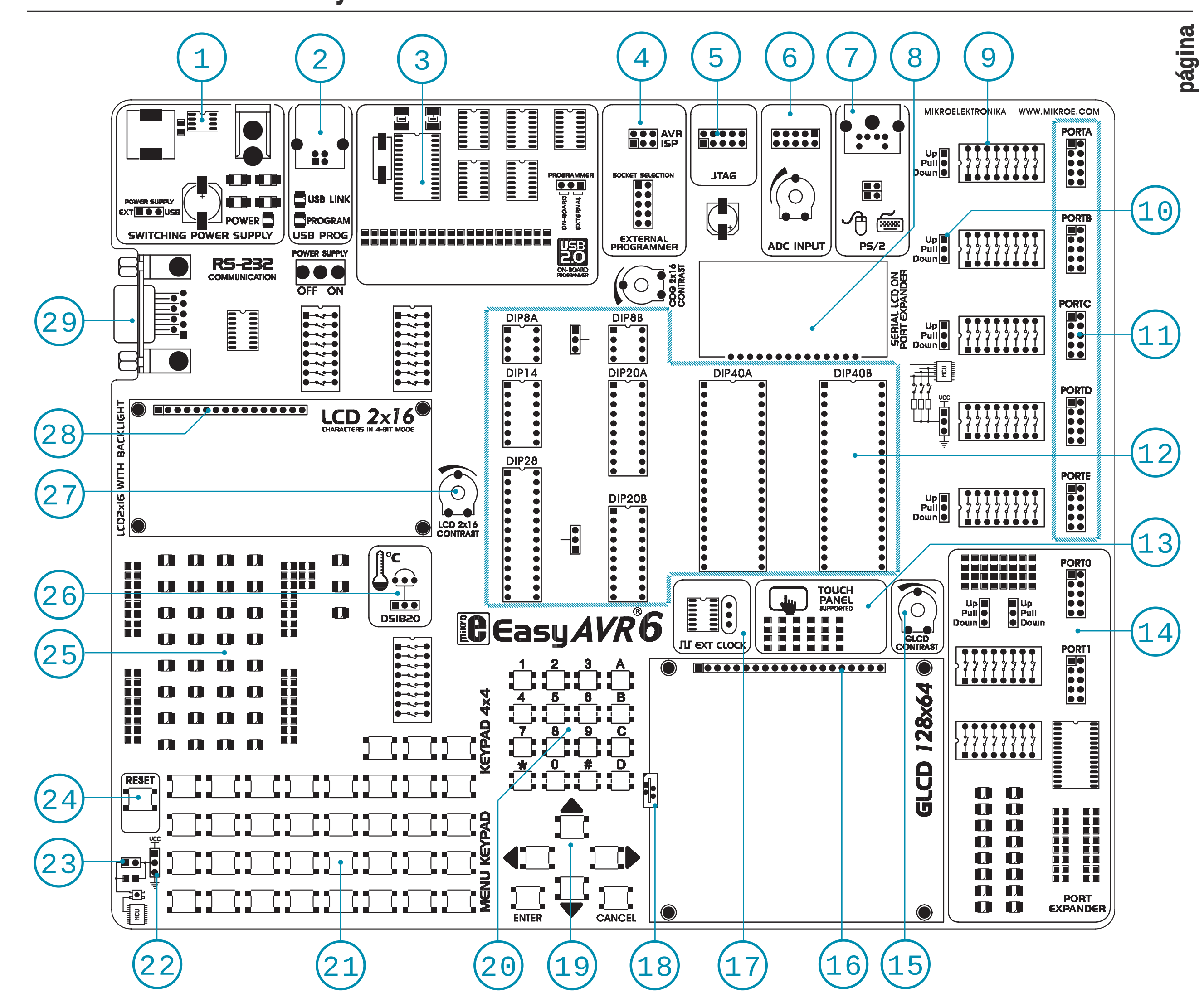
El programa *AVRflash* proporciona una lista completa de todos los microcontroladores soportados. La última versión de este programa con la lista actualizada de los microcontroladores soportados se puede descargar desde nuestra página web: **www.mikroe.com**

El paquete contiene:

Placa de desarrollo :	*EasyAVR6*
CD:	CD del producto con el software apropiado
Cables:	cable USB
Documentación:	manueles *EasyAVR6* y *AVRflash*, guía rápida *Instalación de los controladores USB drivers* y *Esquema eléctrico* del sistema de desarrollo *EasyAVR6*

Especificación del sistema:

Fuente de alimentación:	por el conector DC (7 -23V AC o 9-32V DC) o por el cable USB (5V DC)
Consumo de corriente:	50mA en estado inactivo (los módulos incorporados están inactivos)
Tamaño:	26,5 x 22cm (10,4 x 8,6inch)
Peso:	~417g (0.92lbs)



Prestaciones principales

1. Regulador del voltaje de la fuente de alimentación
2. Conector USB del programador integrado en la placa
3. Programador USB 2.0 integrado en la placa *AVRprog*
4. Conector del programador externo *AVRISP*®
5 Conector de interfaz *JTAG*®
6. Entradas de prueba del conversor A/D
7. Conector PS/2
8. Visualizador LCD integrado en la placa de 2x16 caracteres
9. Interruptores DIP que permiten el funcionamiento de los resistores pull-up/pull-down
10. Selección del modo pull-up/pull-down
11. Conectores de los puertos de E/S
12. Zócalos para inserción de micorocntroladores AVR
13. Controlador del panel táctil
14. Expansor de puertos (port expander)
15. Potenciómetro de contraste del visualizador gráfico LCD de 128x64 caracteres
16. Conector del visualizador gráfico LCD de 128x64 caracteres
17. Oscilador de reloj
18. Conectpor de panel táctil
19. Teclado *MENU*
20. Teclado 4x4
21. Botones de presión para simular las entradas digitales
22 Selector del estado lógico
23. Puente para poner en cortocircuito el resistor protector
24. Botón para resetear el microcontrolador
25. 35 diodos LED para indicar el estado lógico de los pines
26. Zócalo para el sensor de temperatura DS1820
27 Ajuste de contraste del visualizador LCD alfanumérico
28. Conector del visualizador LCD alfanumérico
29. Conector para comunicación RS-232

1.0.Conexión del sistema al PC

Paso 1:

Siga las instrucciones para instalación de los controladores USB y del programador *ACRflash* proporcionadas en los manuales relevantes. No es posible programar los microcontroladores sin haber instalado estos dispositivos anterirormente. En caso de que Ud. ya tenga algún compilador de MikroElektronika instalado en el PC, no es necesario reinstalar el programador *AVRflash* ya que se instala automáticamente al instalar el compilador.

Paso 2:

La conexión del sistema *EasyAVR6* al PC se realiza por medio del cable USB. Una punta del cable USB proporcionado con el conector de tipo USB **B** se conecta al sistema de desarrollo como se muestra en la Figura 1-2. La otra punta del cable (de tipo USB **A**) se conecta al PC. Al establecer la conexión, asegúrese de que el puente J6 se coloque en la posición USB como se muestra en la Figura 1-1.

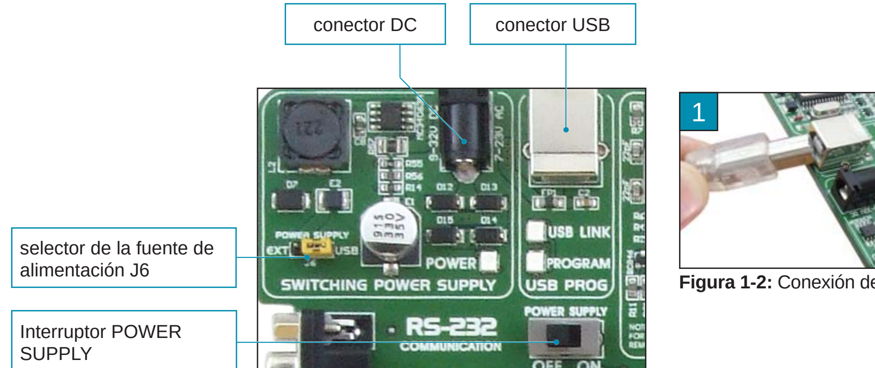


Figura 1-1: Fuente de alimentación

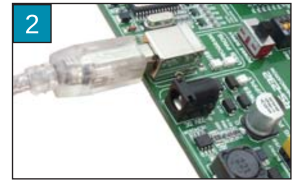


Figura 1-2: Conexión del cable USB (puente J6 en la posición USB)

Paso 3:

Encienda el sistema de desarrollo al poner el interruptor de encendido en la posición ON. Se encenderán dos diodos LED etiquetados como "POWER" y "USB LINK" para indicar que el sistema de desarrollo está listo para su uso. Utilice el programador incorporado *AVRprog y* el pprograma *AVRflash* para volcar el código en el microcontrolador. Después de hacerlo, utilice la placa para probar y desarrollar sus proyectos.

NOTA: Si utiliza algunos módulos adicionales, tales como LCD, GLCD, placas adicionales etc, es necesario colocarlos apropiadamente en el sistema de desarrollo antes de encenderlo. De lo contrario, pueden quedarse dañados permanentemente. Refiérase a la Figura 1-3 para colocar los módulos apropiadamente.

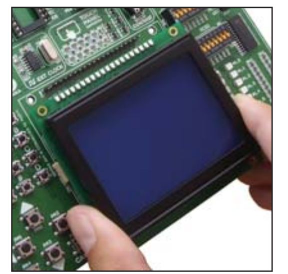
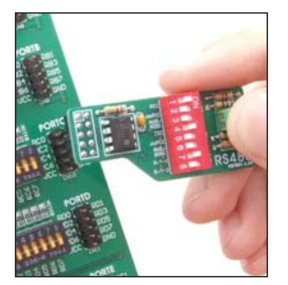




Figura 1-3: Colocación de los módulos adicionales en la placa

2.0. Microcontroladores soportados

El sistema de desarrollo *EasyAVR6* dispone de ocho zócalos separados para inserción de microcontroladores AVR en los encapsulados DIP40, DIP28, DIP20, DIP14 y DIP8. Estos zócalos permiten que los dispositivos soportados en los encapsulados DIP estén conectados directamente a la placa de desarrollo.

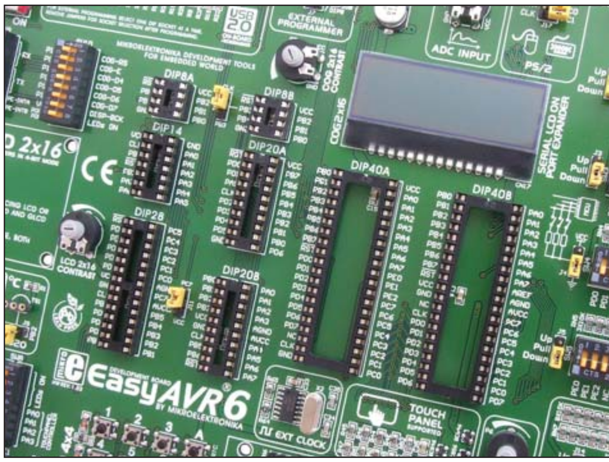


Figura 2-1: Zócalos del microcontrolador

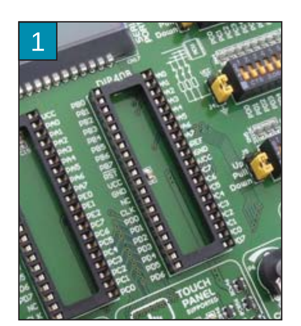
Hay dos zócalos para los microcontroladores AVR en los encapsulados DIP40, DIP20 y DIP8. El zócalo a utilizar depende únicamente de la disposición de los pines en el microcontrolador utilizado. El sistema de desarrollo *EasyAVR6* viene con el microcontrolador en el encapsulado DIP 40.

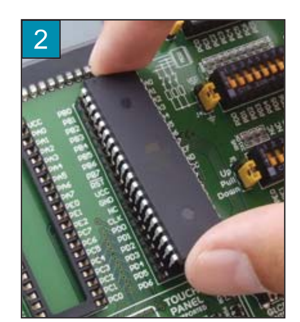
Los puentes J10 and J11 junto a los zócalos DIP28 y DIP8 se utilizan para seleccionar las funciones de los pines del microcontrolador:

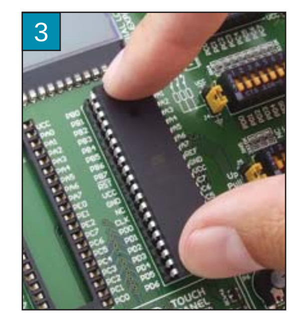
Puente	Posición	Función
J10	PB3	PB3 es un pin de E/S
	OSC	Una señal de reloj del oscilador incorporado se lleva al pin PB3.
J11	VCC	Pin está conectado a VCC
	PC7	PC7 es un pin de E/S

Los microcontroladores AVR pueden utilizar el oscilador incorporado (interno) o el oscilador integrado en la placa (exterior) como una fuente de señal de reloj. El oscilador de reloj integrado en la placa genera las señales de reloj para la mayoría de los microcontroladores soportados.

- Microcontroladores insertados en el zócalo DIP8A utilizan el oscilador interno para la generación de reloj y no están conectados al oscilador externo.
- Microcontroladores insertados en el zócalo DIP8B pueden utilizar tanto el oscilador interno como el oscilador externo, lo que depende de la posición del puente J10.







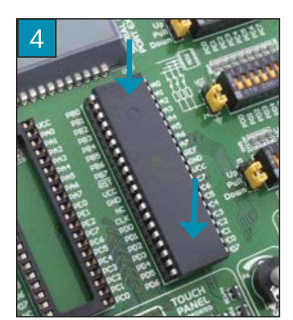


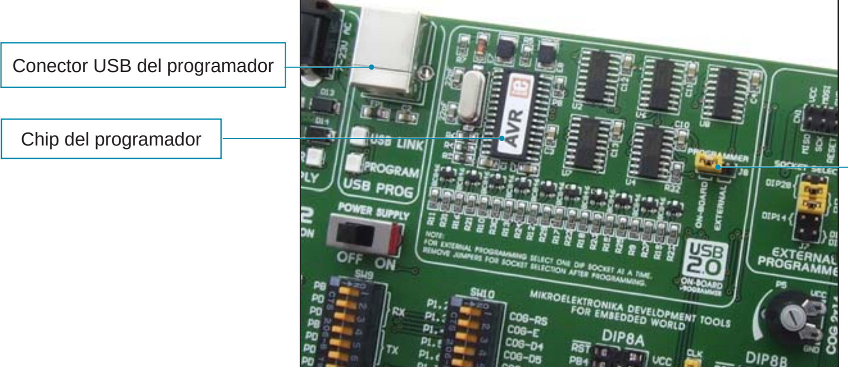
Figura 2-2: Inserción del microcontrolador en el zócalo apropiado

Antes de colocar el microcontrolador en el zócalo adecuado, asegúrese de que la fuente de alimentación esté apagada. En la Figura 2.2. se muestra cómo colocar correctamente un microcontrolador. La Figura 1 muestra el zócalo DIP 40 vacío. Coloque una punta del microcontrolador en el zócalo como se muestra en la Figura 2. Entonces ponga lentamente el microcontrolador más abajo hasta que los pines encajen en el zócalo, como se muestra en la Figura 3. Compruebe una vez más si todo está colocado correctamente y presione el microcontrolador lentamente hasta que encaje en el zócalo completamente, como se muestra en la Figura 4.

NOTA: En la placa de desarrollo se puede colocar un sólo microcontrolador.

3.0. Programador USB 2.0 integrado en la placa AVRprog

El programador *AVRprog* es una herramienta utilizada para volcar el código .hex en el microcontrolador. El sistema de desarrollo *EasyAVR* dispone del programador incorporado en la placa *AVRprog* que permite establecer una conexión entre el microcontrolador y el PC. La figura 3-2 muestra la conexión entre el compilador, el programa *AVRflash* y el microcontrolador.



Puente J8 utilizado para seleccionar el programador (interno o externo) que va a ser utilizado para la programación del chip AVR

Figura 3-1: programador *AVRprog*

mikroC PRO for AVR - C:\Documents and Settings

File Edit View Project Run Tools Help

Lcd.c

```
char txt1[] = "mikroElektronika
char txt2[] = "EasyAVR6";
char txt3[] = "Lcd4bit";
char txt4[] = "example";

char i;

void Move_Delay() {
  Delay_ms(5000);
}
```

Compiling program

1110001001 Bin.
0110100011
0111 2FC23AA7
1011 F43E0021A
Hex. DA67F0541

Loading HEX code

MCU

① Escriba el programa en alguno de los compiladores AVR y genere un fichero .HEX;

② Utilice el programa *AVRlash* para seleccionar un microcontrolador apropiado y para cargar el fichero .HEX;

③ Haga click en el botón *Write* para programar el microcontrolador.

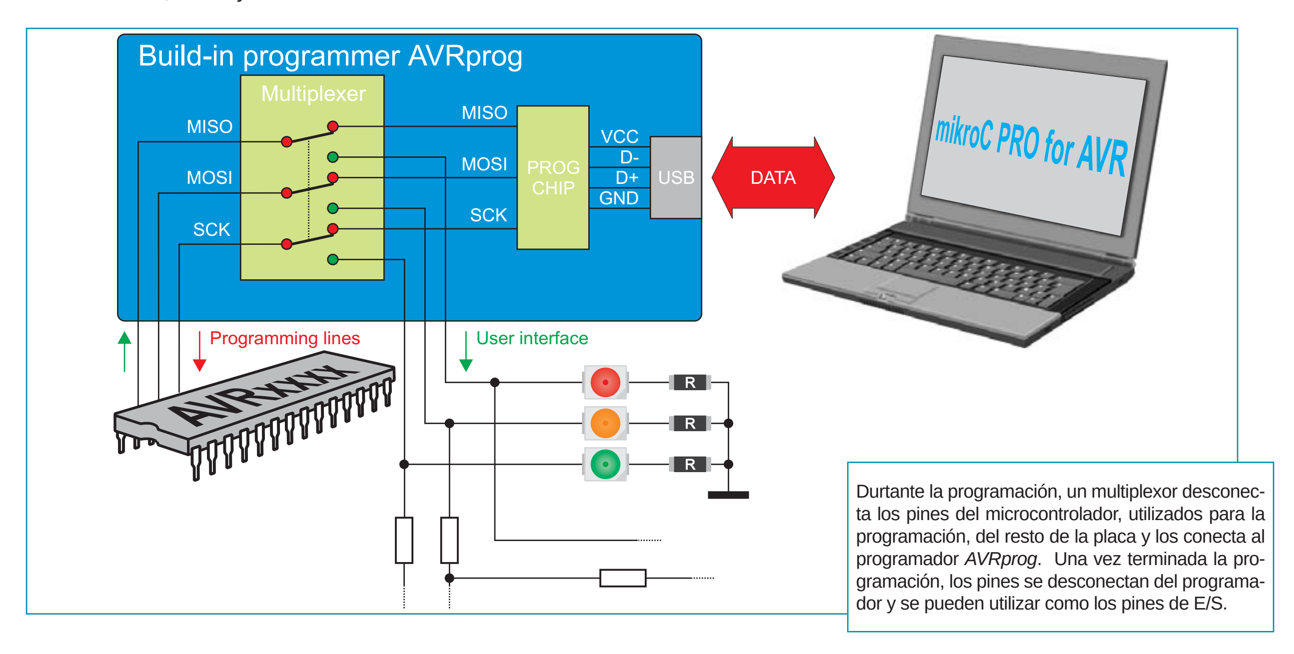
Escriba un código en alguno de los compiladores AVR y genere un fichero .hex. El programador integrado en la placa se encargará de cargar los datos en el microcontrolador.

En el lado derecho de la ventana principal de programa *AVRflash* se encuentran los numerosos botones que facilitan el proceso de programación. En el fondo de la ventana se encuentra una opción que permite monitorizar el progreso de la programación.

Figura 3-2: Principio de funcionamiento del programador

NOTA: Para obtener más informaciones del programador *AVRprog* refiérase al manual relevante proporcionado con el paquete del sistema de desarrollo *EasyAVR6*.

Los microcontroladores AVR se programan por medio de la comunicación serial SPI utilizando los siguientes pines del microcontrolador: MISO, MOSI y SCK.



Durtante la programación, un multiplexor desconecta los pines del microcontrolador, utilizados para la programación, del resto de la placa y los conecta al programador *AVRprog*. Una vez terminada la programación, los pines se desconectan del programador y se pueden utilizar como los pines de E/S.

4.0. Programador externo AVRISP mkII

Además del programador integrado en la placa, el sistema de desarrollo *EasyAVR6* puede utilizar también el programador externo *AVRISP* de la compañía *Atmel* para la programación de los microcontroladores. Este programador está insertado en el conector *AVRISP.*
Para programar un microcontrolador, es necesario poner el puente J8 en la posición EXTERNAL antes de encender el programador. Entonces utilice el puente J7 para seleccionar el zócalo del microcontrolador apropiado.



Figura 4-1: Configurar el puente J7



Puente J8 en la posición EXTERNAL habilita el programador externo *AVRISP*



Puente J8 en la posición ON-BOARD habilita el programador integrado en la placa.

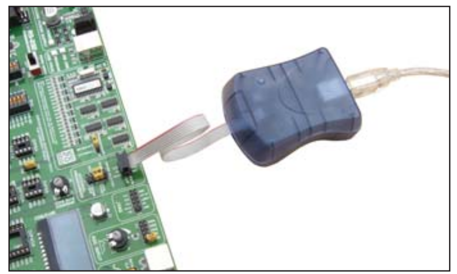
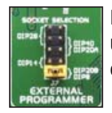


Figura 4-2: *AVRISP mkII* conectado al sistema de desarrollo



Posición del puente J7 cuando el programador externo se utilice para la programación de los microcontroladores en los encapsulados DIP20B y DIP8.



Posición del puente J7 cuando el programador externo se utilice para la programación de los microcontroladores en el encapsulado DIP14.



Posición del puente J7 cuando el programador externo se utilice para la programación de los microcontroladores en los encapsulados DIP40 y DIP20A.



Posición del puente J7 cuando el programador externo se utilice para la programación de los microcontroladores en el encapsulado DIP28.

5.0.Conector JTAG

JTAG ICE es un emulador utilizado para los microcontroladores AVR con la interfaz incorporada *JTAG* (microcontroladores Mega AVR). Antes que nada, *JTAG ICE* está destinado a trabajar con el programa *AVR Studio.* El conector *JTAG* incorporado en los microcontroladores AVR es una versión modificada de la interfaz original *JTAG*. Habilita cambiar el contenido de las memorias internas EEPROM y FLASH (programación de los microcontroladores).



Figura 5-1: Conector *JTAG*

Figura 5-2: *JTAGICE mkII* conectado al sistema de desarrollo

El conector *JTAG* está directamente conectado a los pines del microcontrolador así que no depende de la configuración de los puentes J7 y J8. De contrario esta conexión debe ser realizada utilizando los programadores *AVRprog* y *AVRISP.*

6.0. Oscilador de reloj

Hay un oscilador de reloj proporcionado en la placa utilizado como una fuente externa de señal de reloj. El cristal de cuarzo utilizado con el propósito de estabilizar la frecuencia de reloj está insertado en el zócalo apropiado y por eso siempre se puede reemplazar por otro. Su valor máximo depende de la frecuancia de funcionamiento máxima del microcontrolador.

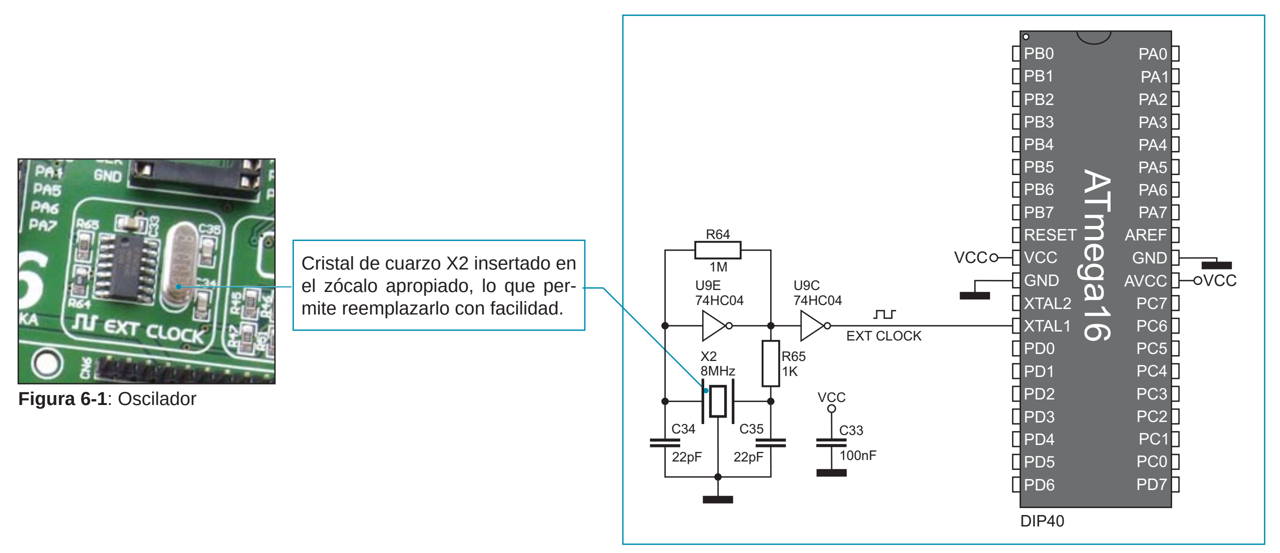


Figura 6-1: Oscilador

Figura 6-2: Esquema de conexión del oscilador

7.0. Fuente de alimentación

El sistema de desarrollo *EasyAVR6* puede utilizar una de dos fuentes de alimentación:

1. Fuente de alimentación de +5V desde el PC por el cable USB de programación;
2. Fuente de alimentación externa conectada a un conector DC en la placa de desarrollo.

El regulador del voltaje MC34063A y el rectificador Gretz permiten que el voltaje de la fuente de alimentación externa sea AC (en el rango de 7V a 23V) o DC (en el rango de 9V a 32V). El puente J6 se utiliza como selector de la fuente de alimentación. Cuando se utilice la fuente de alimentación USB, el puente J6 debe estar en la posición USB. Cuando se utilice la fuente de alimentación externa, el puente J6 debe estar en la posición EXT. El sistema de desarrollo se enciende/apaga al poner el interruptor POWER SUPPLY en la posición ON.

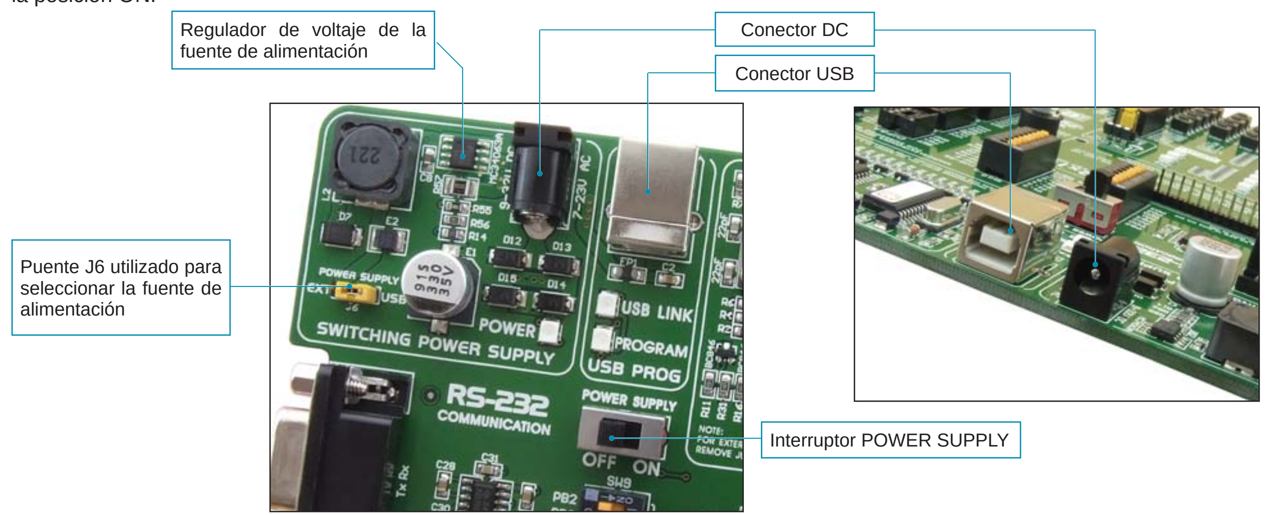


Figura 7-1: Fuente de alimentación

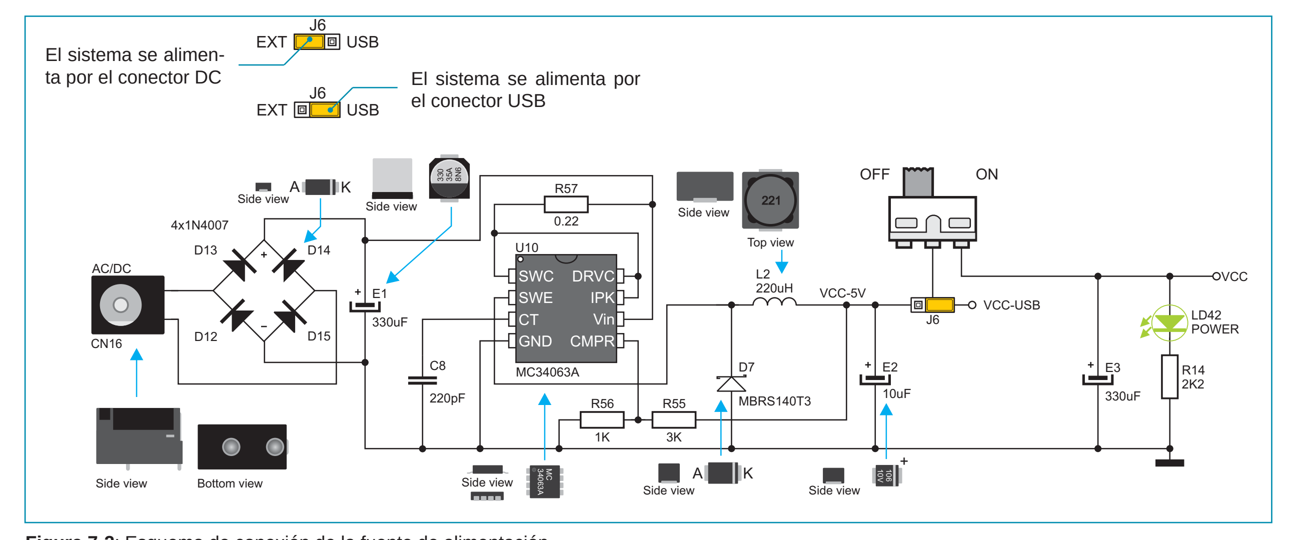


Figura 7-2: Esquema de conexión de la fuente de alimentación

8.0. Interfaz de comunicación RS-232

USART (universal synchronous/asynchronous receiver/transmitter) - transmisor/receptor asíncrono/universal es una de las formas más frecuentes de intercambiar los datos entre el PC y los periféricos. La comunicación serial **RS-232** se realiza por medio de un conector SUB-D de 9 pines y el módulo USART del microcontrolador. Para habilitar esta comunicación, es necesario establecer una conexión entre las líneas de comunicación RX y TX y los pines del microcontrolador, que están conectados con el módulo USART, utilizando un interruptor DIP SW9. Los pines del microcontrolador utilizados en esta comunicación serial están marcados de la siguiente manera: RX (*receive data)* - *recibir datos* y TX (*transmit data)* - *transmitir datos.* La velocidad de transmisión en baudios es hasta 115 kbps. Para habilitar que el módulo USART del microcontrolador reciba las señales de entrada con diferentes niveles de voltaje, es necesario proporcionar un convertidor de nivel de voltaje como MAX-202C.

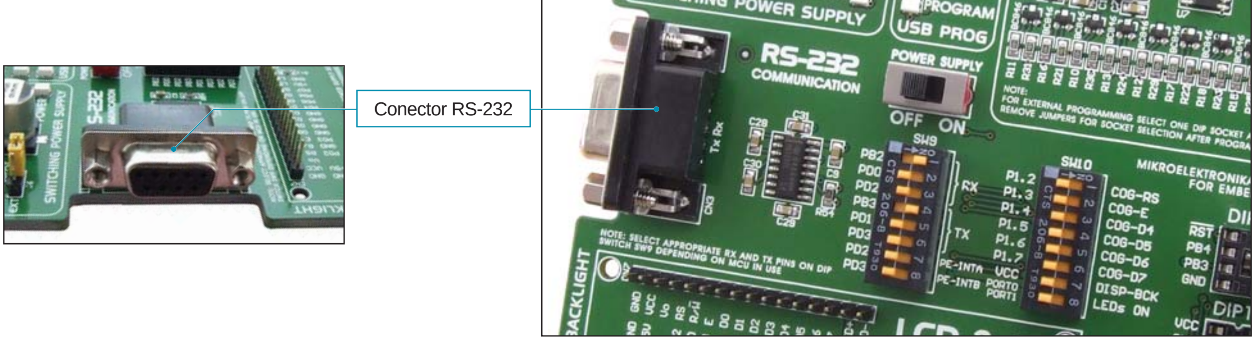


Figura 8-1: módulo RS-232

La función de los interruptores DIP SW7 y SW8 es de determinar cuál de los pines de microcontrolador se utilizará como líneas RX y TX. La disposición de los pines difiere dependiendo del tipo de microcontrolador. La Figura 8-2 muestra la conexión entre el módulo RS-232 y el microcontrolador en el encapsulado DIP40 (*ATMEGA16*).

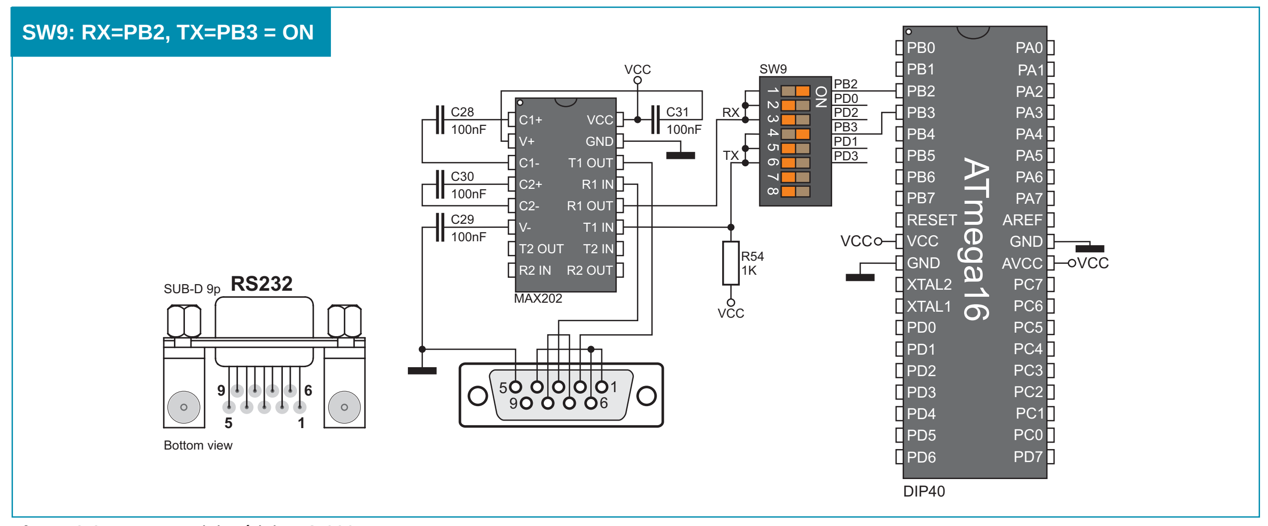


Figure 8-2: Esquema del módulo RS-232

NOTE: Asegúrese de que su microcontrolador esté proporcionado con el módulo USART, ya que no está necesariamente integrado en todos los microcontroladores AVR.

9.0. Interfaz de comunicación PS/2

El conector **PS/2** permite conectar los dispositivos de entrada, tales como teclado y ratón, con el sistema de desarrollo. Para habilitar la comunicación PS/2 es necesario colocar apropiadamente los puentes J16 y J17, así que las líneas DATA y CLK están conectadas a los pines del microcontrolador PC0 y PC1. No conecte/desconecte las unidades de entrada al conector PS/2 mientras que el sistema de desarrollo está encendido debido a que puede dañar el microcontrolador permanentemente.

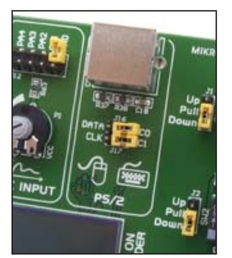


Figura 9-1: Conector PS/2 (J16 y J17 no están colocados)

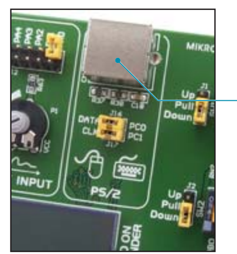
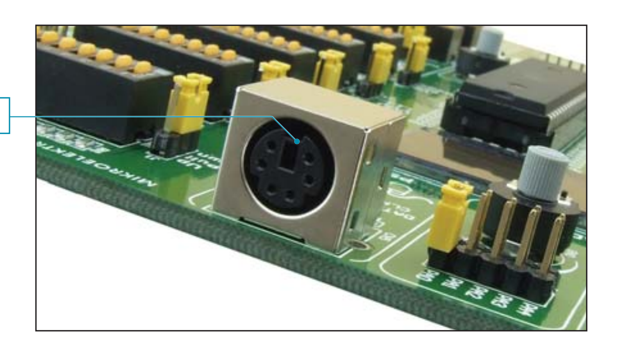


Figura 9-2: Conector PS/2 (J16 y J17 están colocados)

Conector PS/2



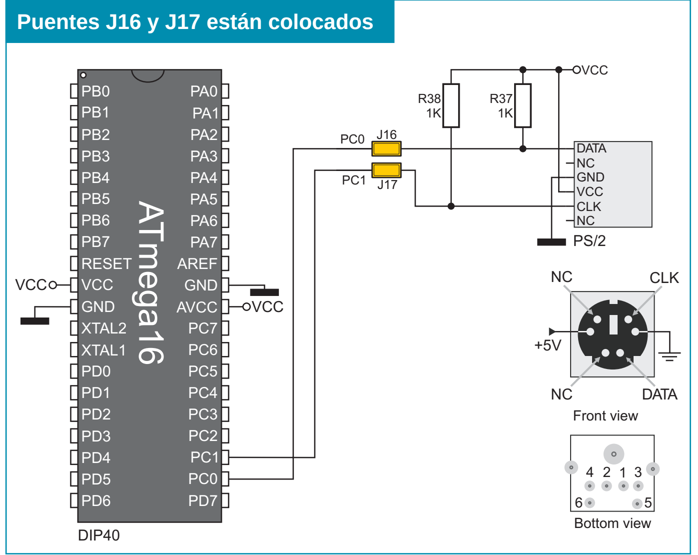


Figura 9-3: Esquema de conexión del conector PS/2



Figura 9-4: *EasyAVR6* conectado al teclado

10.0. Sensor de temperatura DS1820

La comunicación serial **1-wire**® permite la transmisión de datos por medio de una sóla línea de comunicación, mientras que el proceso mismo está bajo el control del microcontrolador maestro. La ventaja de tal comunicación es que se utiliza sólo un pin de microcontrolador. Todos los dispositivos esclavos disponen de un código ID único por defecto, lo que permite que el dispositivo maestro identifique fácilmente los dispositivos que comparten el mismo interfaz.

DS1820 es un sensor de temperatura que utiliza el estándar 1-wire para su funcionamiento. Es capaz de medir las temperaturas dentro del rango de -55 a 125°C y proporcionar la exactitud de medición de ±0.5°C para las temperaturas dentro del rango de -10 a 85°C. Para su funcionamiento el DS1820 requiere un voltaje de la fuente de alimentación de 3 a 5.5V. El DS1820 tarda como máximo 750 ms en calcular la temperatura con una resolución de 9 bits. El sistema de desarrollo *EasyAVR6* proporciona un zócalo separado para el DS1820. Puede utilizar uno de los pines PA4 o PB2 para la comunicación con el microcontrolador. El próposito del puente J9 es de seleccionar el pin que será utilizado para la comunicación 1-wire. En la Figura 10-4. se muestra la comunicación 1-wire con el microcontrolador por el pin PA4.

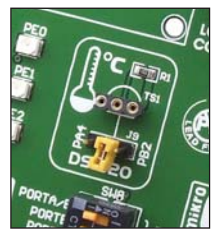


Figura 10-1: Conector DS1820 (no se utiliza la comunicación 1-wire)



Figura 10-2: Puente J11 en la posición a la izquierda (comunicación 1-wire por el pin PA4)

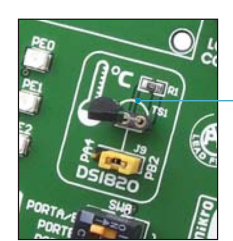


Figura 10-3: Puente J11 en la posición a la derecha (comunicación 1-wire por el pin PB2)

NOTA: Asegúrese de que el semicírculo en la placa coincida con la parte redonda del DS1820.

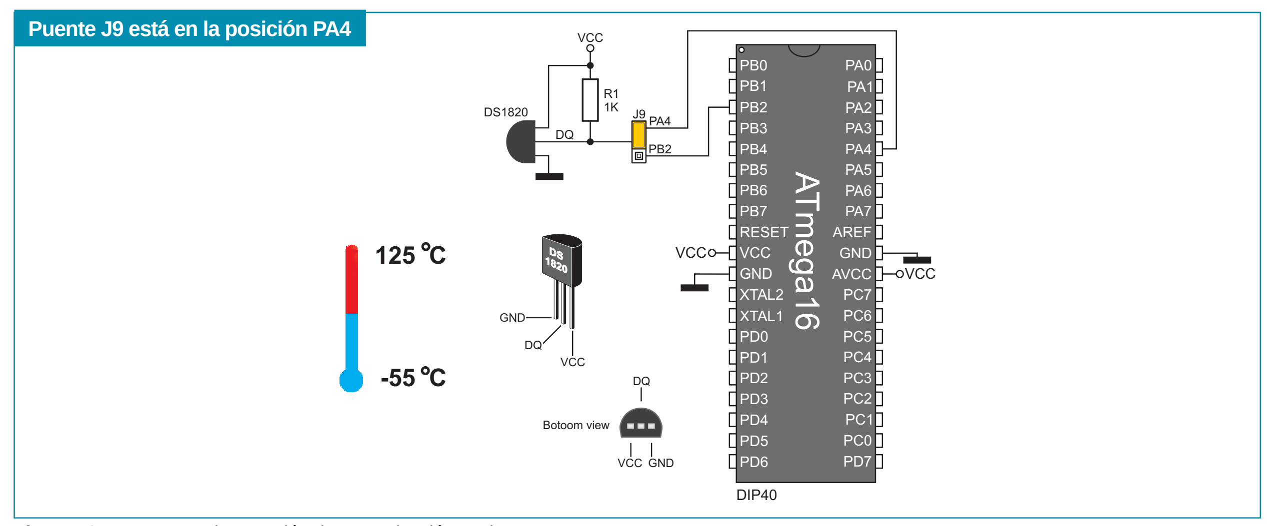


Figura 10-4: Esquema de conexión de comunicación 1-wire

11.0. Entradas de prueba del conversor A/D

Un conversor A/D (analógico-digital) se utiliza con el propósito de covertir una señal analógica a un valor digital apropiado. El conversor A/D es lineal, lo que quiere decir que el número convertido es linealmente dependiente del valor del voltaje de entrada.
El conversor A/D dentro del microcontrolador convierte un valor de voltaje analógico a un número de 10 bits. Por las entradas de prueba del conversor A/D se pueden llevar los voltajes que varían de 0 a 5V a los microcontroladores con el conversor A/D incorporado.
El puente J12 se utiliza para seleccionar uno de los siguientes pines para la conversión AD: PA0, PA1, PA2, PA3 or PA4. El resistor R63 tiene la función de protección. Se utiliza para limitar el flujo de corriente por el potenciómetro o por el pin del microcontrolador. El valor del voltaje analógico de entrada se puede cambiar linealmente utilizando el potenciómetro P1 (10k).



Figura 11-1: ADC (posición por defecto del puente)



Figura 11-2: El pin PA0 utilizado como el pin de entrada para la conversión A/D

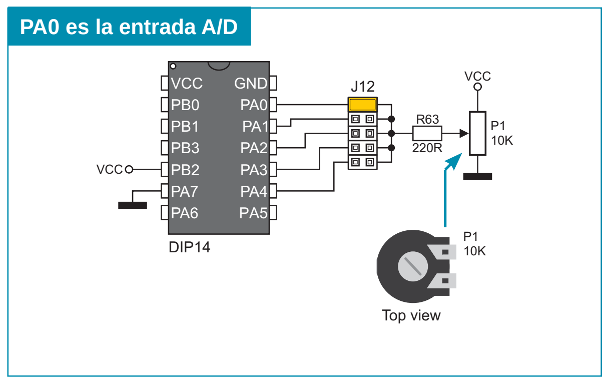


Figura 11-3: Conexión del microcontrolador AVR en el encapsulado DIP14 a las entradas de prueba del conversor A/D.

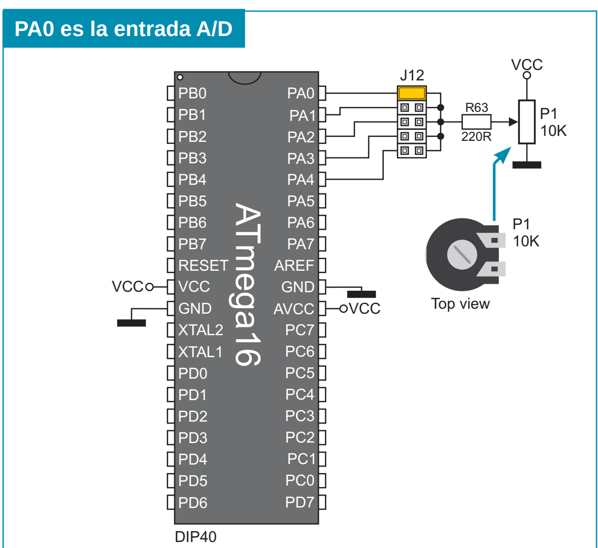


Figure 11-4: Conexión del microcontrolador en el encapsulado DIP40 a las entradas de prueba del conversor A/D.

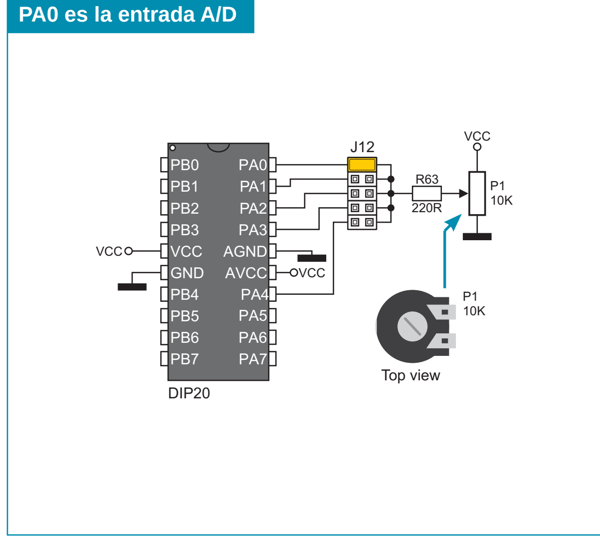


Figura 11-5: Conexión del microcontrolador en el encapsulado DIP20B a las entradas de prueba del conversor A/D.

NOTA: Para que un microcontrolador pueda realizar una conversión A/D con exactitud, es necesario apagar los diodos LED y los resistores pull-up/pull-down en los pines de los puertos utilizados por el conversor A/D.

12.0. Diodos LED

El diodo LED (Light-Emitting Diode) - Diodo emisor de luz, representa una fuente electrónica de luz de muy alta eficacia. Al conectar los LEDs es necesario colocar el resistor para limitar la corriente cuyo valor se calcula utilizando la fórmula R=U/I donde R se refierre a la resistencia expresada en ohmios, U se refierre al voltaje en el LED, I se refierre a la corriente del LED. El voltaje común del LED es aproximadamente 2.5V, mientras que la intensidad de corriente varía de 1 a 20mA dependiendo del tipo del diodo LED. El sistema de desarrollo *EasyAVR6* utiliza los LEDs con la corriente I=1mA.

El sistema de desarrollo *EasyAVR6* tiene 35 LEDs que indican visualmente el estado lógico en cada pin de E/S del microcontrolador. Un diodo LED activo indica la presencia de un uno lógico (1) en el pin. Para habilitar que se muestre el estado de los pines, es necesario seleccionar el puerto apropiado PORTA/E, PORTB, PORTC o PORTD utilizando el interruptor DIP SW8.

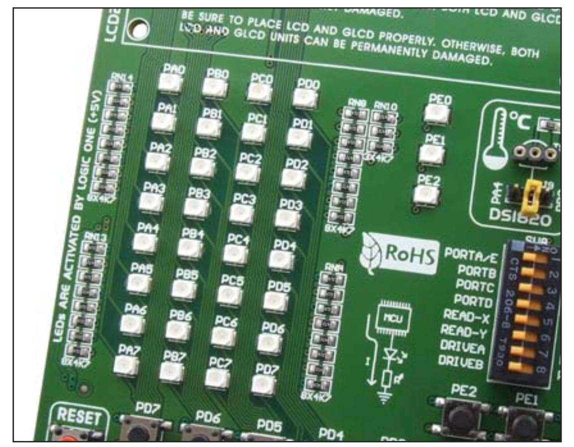


Figure 12-1: LEDs

Muesca indica el cátodo SMD LED

A K

PA0 PA1 PA2 PA3 PA4

I R=U/I

A K SMD LED 472 R

Microcontrolador

Resistor SMD limita el flujo de corriente por un LED

SW8: PORTA = ON

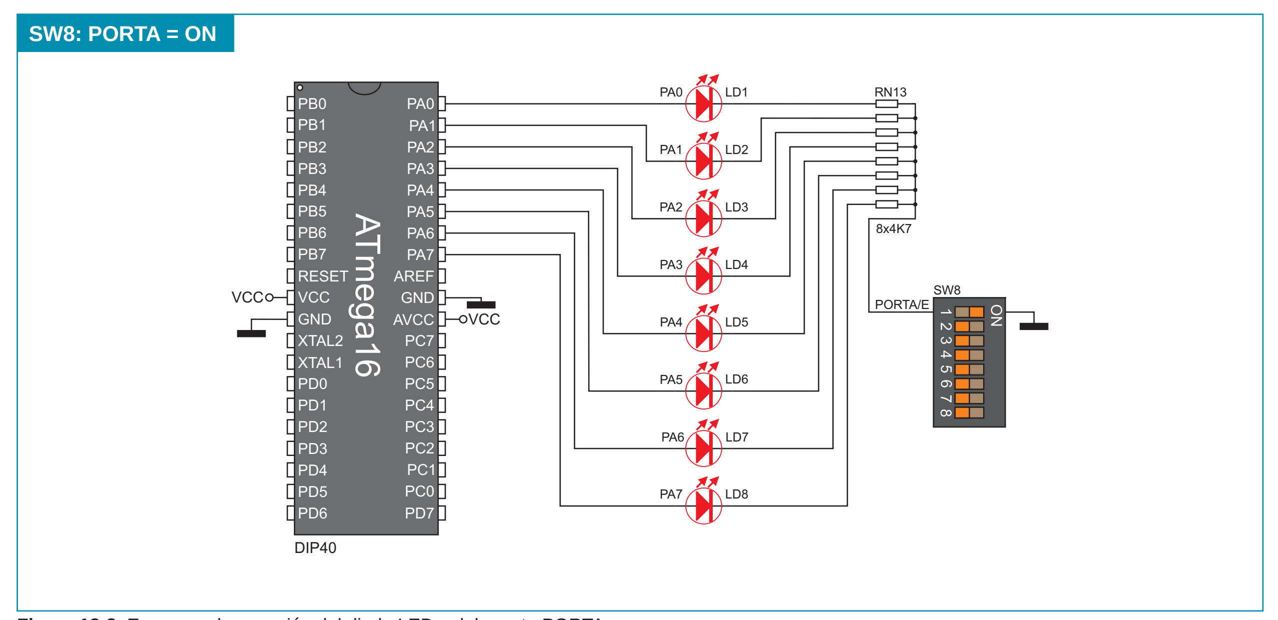


Figure 12-2: Esquema de conexión del diodo LED y del puerto PORTA

13.0. Botones de presión

El estado lógico de todos los pines de entrada digitales del microcontrolador se puede cambiar al utilizar los botones de presión. El puente J13 se utiliza para determinar el estado lógico que será aplicado al pin deseado del microcontrolador al apretar el botón apropiado. El propósito del resistor protector es de limitar la máxima corriente lo que impide la ocurrencia de un corto circuito. Los usuarios con más experiencia pueden, si es necesario, pueden cortocircuitar este resistor utilizando el puente J18. Justamente junto a los botones de presión, se encuentra un botón RESET que no está conectado al pin MCLR. La señal de reset se genera por el programador.

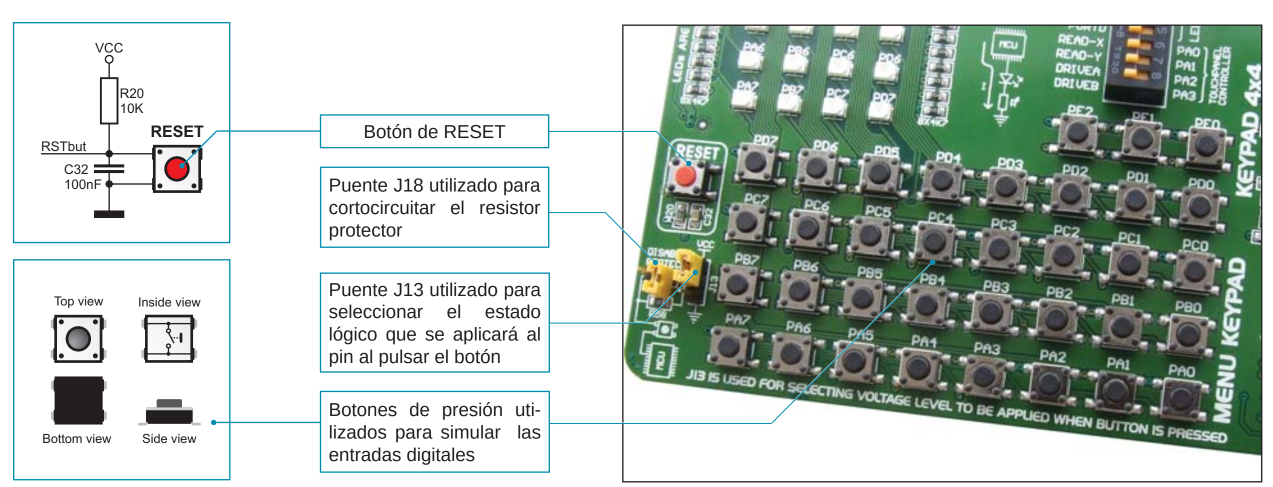


Figura 13-1: Botones de presión utilizados para simular las entradas digitales

Al pulsar cualquier botón de presión (PA0-PA7) cuando el puente J13 se encuentre en la posición VCC, un 1 lógico (5V) será aplicado al pin apropiado del microcontrolador como se muestra en la Figura 13-2.

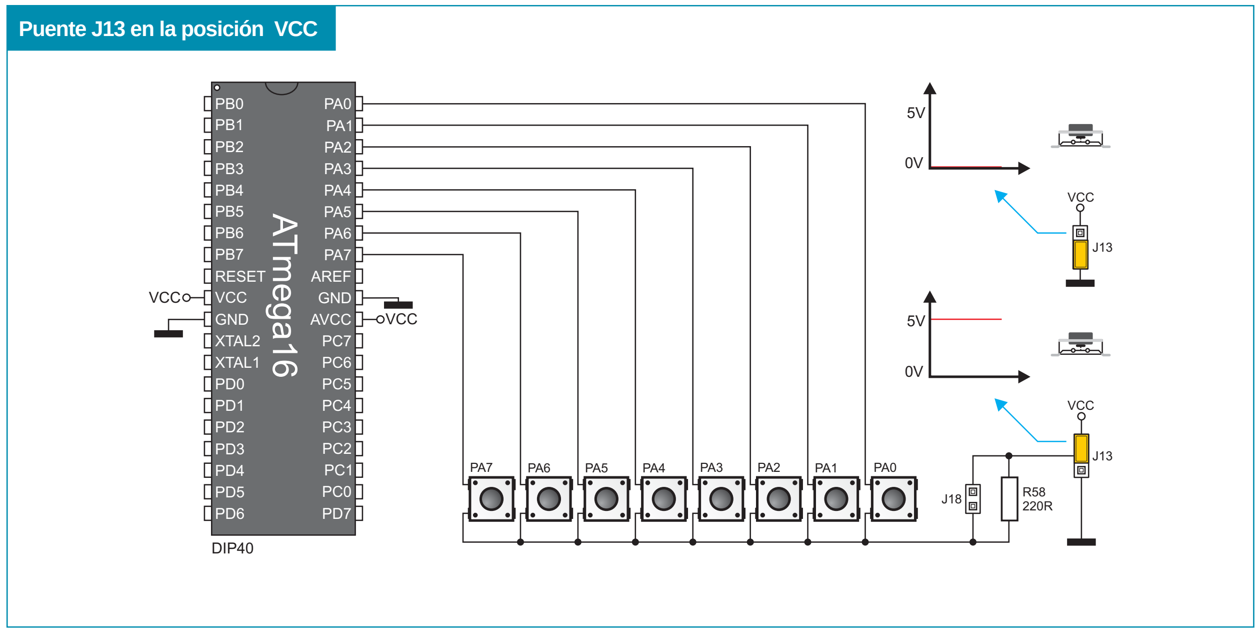


Figure 13-2: Esquema de conexión de los botones de presión al puerto PORTA

14.0. Teclados

En el sistema de desarrollo *EasyAVR6* se encuentran dos teclados. Estos son el teclado 4x4 y el teclado *MENU*. El teclado 4x4 representa un teclado alfanumérico estándar conectado al PORTC del microcontrolador. El funcionamiento de este teclado está basado en el principio de escanear los pines según el que los pines PC0, PC1, PC2 y PC3 se configuran como las entradas conectadas a los resistores *pull-down*. Los pines PC4, PC5, PC6 y PC7 se configuran como las salidas de voltaje da alto nivel. Al presionar cualquier botón un uno lógico (1) se aplicará a los pines de entrada. La detección del botón de presión se realiza por medio de software. Por ejemplo, al presionar el botón de presión '6' un uno lógico (1) aparece en el pin PC2. Para detectar qué botón de presión fue presionado, un uno lógico (1) se aplica a cada pin de entrada PC4, PC5, PC6 y PC7.
Los botones del teclado *MENU* están conectados de la manera similar a los botones en el puerto PORTA. La única diferencia yace en la disposición de los botones. Los botones del teclado *MENU* están colocados de tal manera que porporcionen una navegación fácil por menús.



Figura 14-1: Teclado 4x4

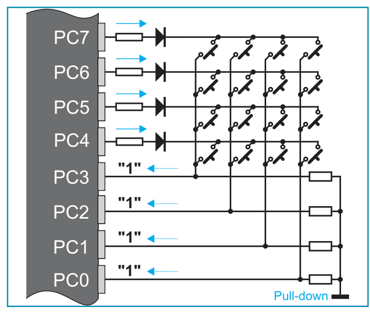


Figura 14-2: Rendimieno del teclado 4x4

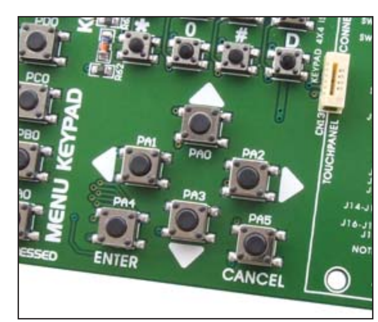


Figura 14-3: Teclado *MENU*

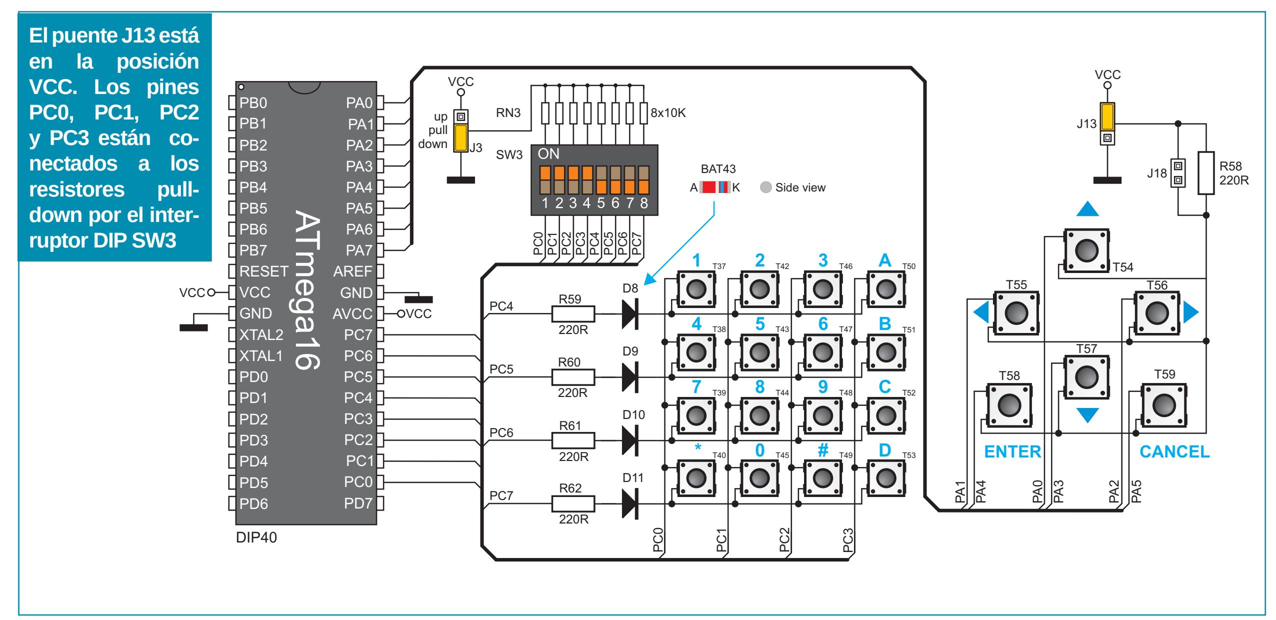


Figura 14-4: Esquema de conexión de los teclados (4x4 y MENU) al microcontrolador

15.0. Visualizador alfanumérico LCD 2x16

El sistema de desarrollo *EasyAVR6* dispone de un conector integrado en la placa en el que se coloca el visualizador alfanumérico LCD 2x16. Este conector está conectado al microcontrolador por el puerto PORTD. El potenciómetro P7 se utiliza para ajustar el contraste del visualizador. El interruptor con la etiqueta *DISP-BCK* en el interruptor DIP SW10 se utiliza para encender/apagar luz de fondo del visualizador.
La comunicación entre un visualizador LCD y el microcontrolador se establece utilizando el modo de 4 bits. Los dígitos alfanuméricos se visualizan en dos líneas de las que cada una contiene hasta 16 caracteres de 7x5 píxeles.



Figura 15-1: Conector del visualizador alfanúmerico LCD 2x16

Conector para la colocación del visualizador alfanumérico LCD

Potenciómetro para ajustar el contraste



Figura 15-2: Visualizador LCD 2x16

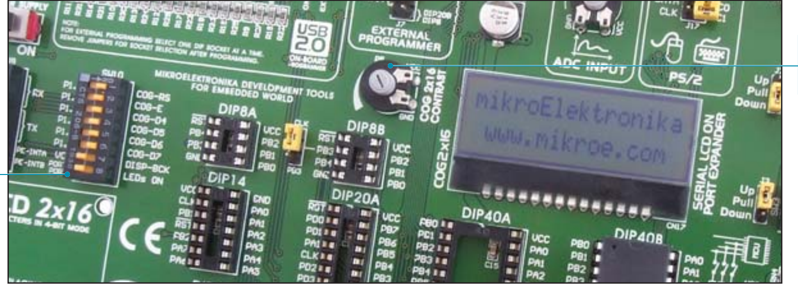
SW10: DISP-BCK = ON

Figura 15-3: Esquema de conexión del visualizador alfanumérico LCD 2x16

16.0. Visualizador LCD incorporado 2x16 con comunicación serial

El visualizador LCD 2x16 incorporado está conectado al microcontrolador por medio del expansor de puertos. Para utilizar este visualizador es necesario poner todos los interruptores (1-6) del interruptor DIP SW10 en la posición ON de modo que el visualizador LCD incorporado se conecte al puerto 1 del expansor de puertos. Los siguientes interruptores DIP SW6, S7 y SW9 permitn que el expansor de puertos utilice la comunicación serial. El potenciómetro P5 se utiliza para ajustar el contraste del visualizador.
A diferencia del visualizador LCD, el visualizador LCD incorporado no dispone de la luz de fondo y recibe los datos por medio del expansor de puertos que utiliza la comunicación SPI para comunicarse con el microcontrolador. Este visualizador también visualiza los dígitos en dos líneas de las que cada una contiene hasta 16 caracteres de 7x5 píxeles.

El interruptor DIP SW10 utilizado para enecender el visualizador LCD incorporado 2x16



Potenciómetro para ajustar el contraste

Figura 16-1: Visualizador LCD incorporado 2x16

SW6, SW7: CS, RST, SCK, MISO, MOSI = ON
SW10: 1-6 = ON

Figura 16-2: Esquema de conexión del visualizador LCD incorporado 2x16

17.0. Visualizador gráfico LCD 128x64

El visualizador gráfico LCD (128x64 GLCD) proporciona un método avanzado de visualizar los mensajes gráficos. Está conectado al microcontrolador por los puertos PORTC y PORTD. El visualizador GLCD dispone de la resolución de pantalla de 128x64 píxeles que permite visualizar diagramas, tablas y otros contenidos gráficos. Puesto que el puerto PORTD también es utilizado por el visualizador alfanumérico LCD 2x16, no es posible utilizar los dos simultáneamente. El potenciómetro P6 se utiliza para ajustar el contraste del visualizador GLCD. El interruptor 7 en el interruptor DIP SW10 se utiliza para encender/apagar la luz de fondo del visualizador.

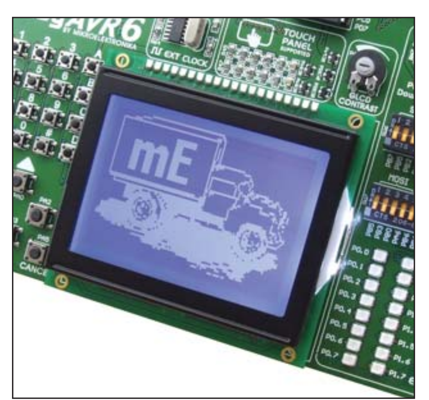


Figura 17-1: Visualizador GLCD

Potenciómetro para ajustar el contraste

Conector GLCD

Conector del panel táctil

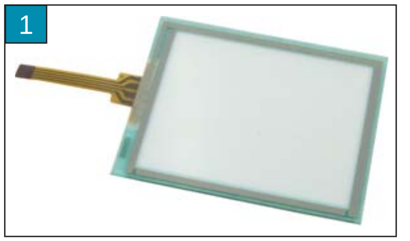
Figura 17-2: Conector GLCD

SW10: DISP-BCK = ON

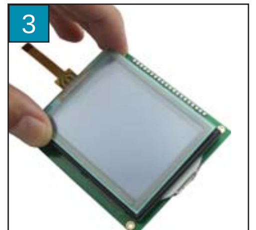
Figura 17-3: Esquema de conexión del visualizador GLCD

18.0. Panel táctil

Un panel táctil es un panel fino, autoadhesivo, transparente y toque senistivo. Se coloca sobre el visualizador GLCD. El propósito principal de este panel es de registrar la presión en un punto específico del visualizador y enviar sus coordenadas en la forma del voltaje analógico al microcontrolador. Los interruptores 5,6,7 y 8 en el interruptor DIPSW9 se utilizan para conectar el panel táctil al microcontrolador.







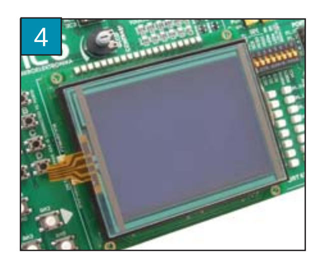


Figura 18-1: Panel táctil

La Figura 18-1 muestra cómo colocar un panel táctil sobre un visualizador GLCD. Asegúrese de que el cable plano esté a la izquierda del visualizador GLCD como se muestra en la Figura 4.

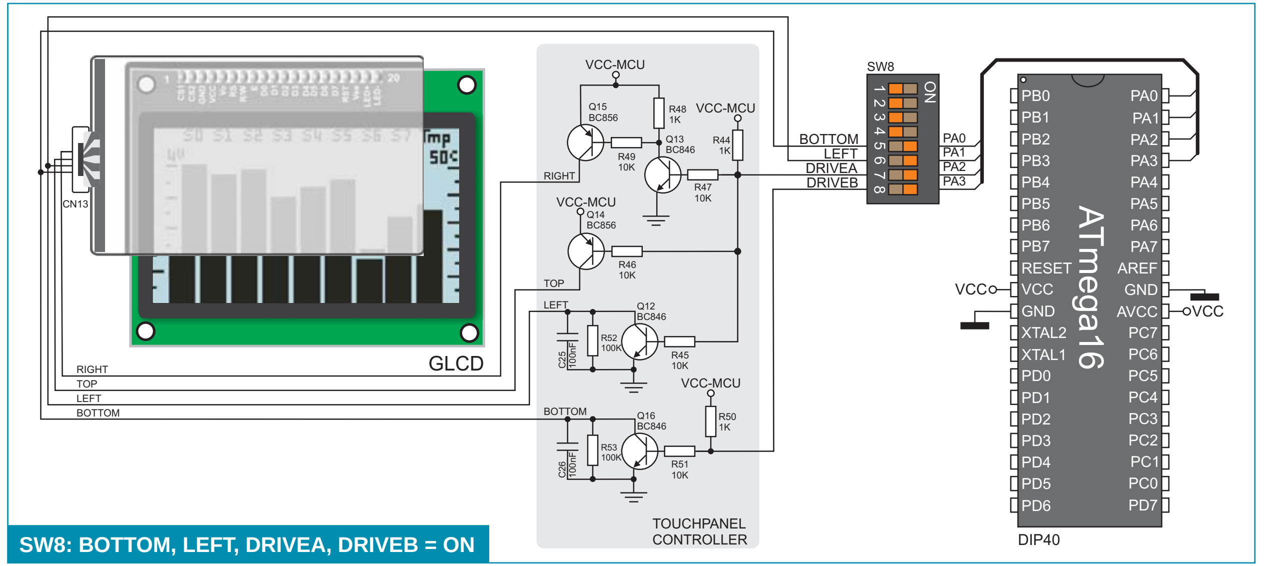
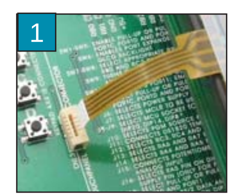
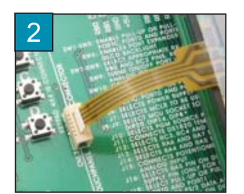
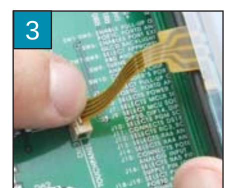


Figura 18-2: Esquema de conexión del panel táctil







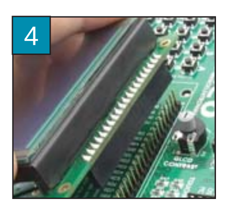


Figure 18-3: Colocación del panel táctil

La Figura 18-3. muestra detalladamente cómo conectar un panel táctil al microcontrolador. Acerque la punta del cable plano al conector CN13 como se muestra en la Figura 1. Inserte el cable en el conector, como se muestra en la Figura 2, y presiónelo lentamente de modo que la punta del cable encaje en el conector completamente, como se muestra en la Figura 3. Luego inserte el visualizador GLCD en el conector apropiado como se muestra en la Figura 4.

NOTA: Los LEDs y los resistores pull-up/pull-down en el puerto PORTA tienen que estar apagados al utilizar un panel táctil.

19.0. Puertos de Entrada/Salida

A lo largo de la parte derecha del sistema de desarrollo están siete conectores de 10 pines que están conectados a los puertos de E/S del microcontrolador. Unos están conectados directamente a los pines del microcontrolador, mientras que otros están conectados por medio de los puentes. Los interruptores DIP SW1-SW5 permiten que cada pin de conector esté conectado a un resistor pull up/pull down. Si los pines de los puertos están conectados a un resistor pull-up o pull-down depende de la posición de los puentes J1-J5.

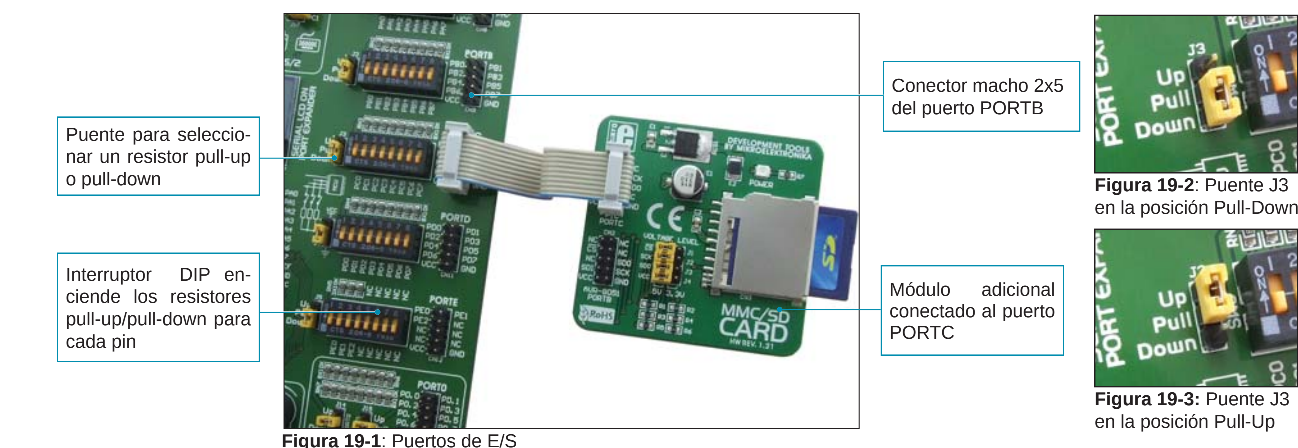


Figura 19-1: Puertos de E/S

Figura 19-2: Puente J3 en la posición Pull-Down

Figura 19-3: Puente J3 en la posición Pull-Up

SW1: 1-8 = OFF
Puente J1 en la posición pull-down
Puente J13 en la posición VCC

Figura 19-4: Esquema de conexión del puerto PORTA

Los resistores pull-up/pull-down permiten determinar el nivel lógico en todos los pines de entrada del microcontrolador cuando estén en el estado inactivo. El nivel lógico depende de la posición pull-up/pull-down del puente. El pin PA0 junto con el resistor DIP SW1 relevante, el puente J1 y el botón de presión PA0 con el puente J13 se utilizan con el próposito de explicar el funcionamiento de los resistores pull-up/pull-down. El principio de su funcionamiento es idéntico para todos los pines del microcontrolador.

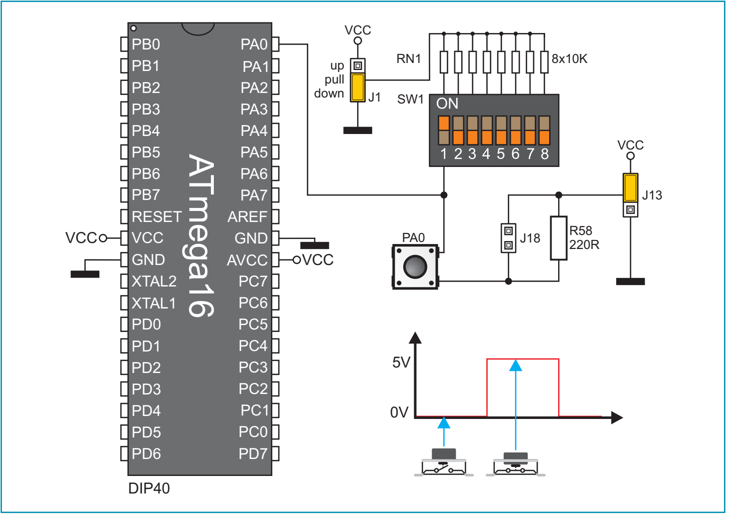


Figura 19-5: Puente J1 en la posición pull-down y el puente J13 en la posición pull-up

Para conectar los pines en el puerto PORTA a los resistores pull-down, es necesario poner el puente J1 en la posición *Down*. Esto permite que se lleve un cero lógico (0) a cualquier pin en el puerto PORTA en el estado inactivo por medio del puente J1 y de la red de resistores de 8x10K. Para llevar esta señal al pin PA0, es necesario poner el interruptor PA0 en el interruptor DIP SW1 en la posición ON.
Por consiguiente, cada vez que se presiona el botón de presión PA0 un uno lógico (1) aparecerá en el pin PA0, con tal de que el puente J13 esté colocado en la posición VCC.

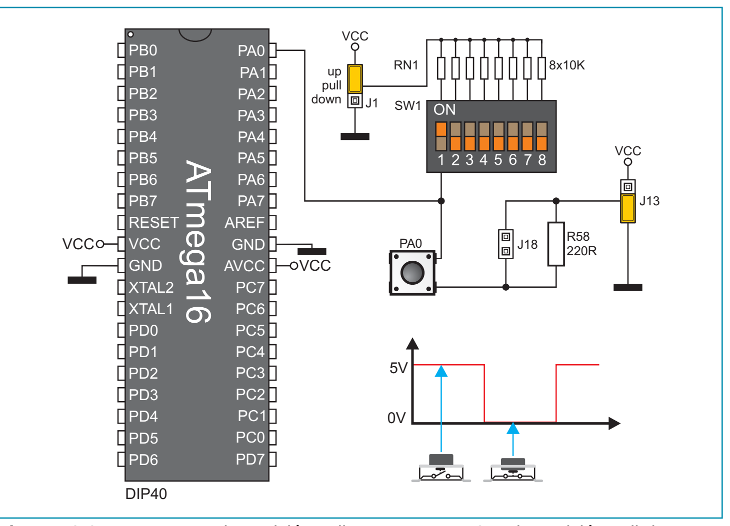


Figura 19-6: Puente J1 en la posición pull-up y puente J13 en la posición pull-down

Para conectar los pines en el puerto PORTA a los resistores pull-up, y para llevar un cero lógico (0) a los pines de entrada en el puerto, es necesario poner el puente J1 en la posición *Up* (5V) y el puente J13 en la posición GND (0V). Además, el pin PA0 en el interruptor DIP SW1 se debe colocar en la posición ON. Esto permite llevar un uno lógico (5V) a todos los pines de entrada del puerto PORTA en el estado inactivo por medio del resistor de 10k. El voltaje se lleva al pin PA0 por medio del resistor de 10k y el interruptor PA0.
Por consiguiente, cada vez que se presiona el botón de presión PA0 un cero lógico (0) aparecerá en el pin PA0.

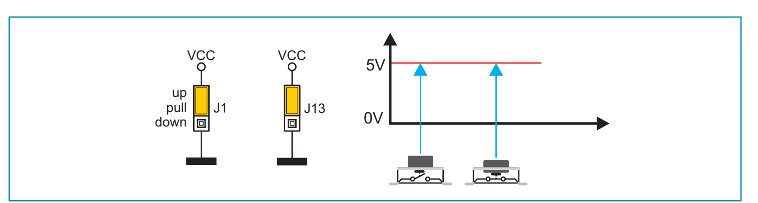


Figura 19-7:. Puentes J1 y J13 en las mismas posiciones

En caso de que los puentes J1 y J13 tengan el mismo estado lógico, al presionar cualquier botón de presión no cambia el estado lógico de los pines de entrada.

20.0. Expansor de puertos (Puertos de Entrada/Salida adicionales)

Las líneas de comunicación SPI y el circuito MCP23S17 proporcionan el sisrema de desarrollo *EasyAVR6* con recursos de incrementar en dos el número de los puertos de E/S disponibles. Si el expansor de puertos se comunica con el microcontrolador por los interruptores DIP SW6 y SW7, los pines del microcontrolador utilizados para la comunicación SPI no se pueden utilizar como los pines de E/S. Los interruptores INTA e INTB en el interruptor DIP SW9 habilitan una interrupción utilizada por el circuito MCP23S17.

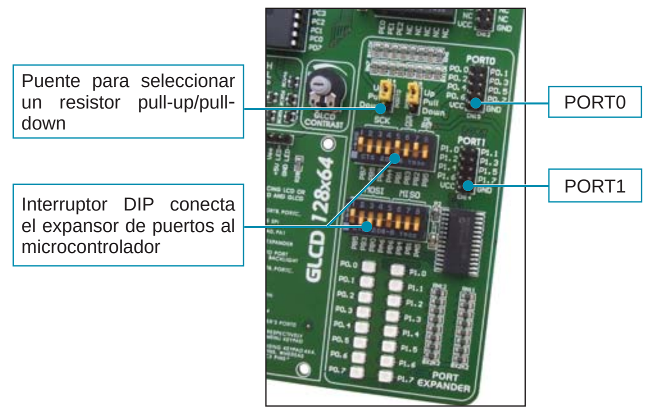


Figura 20-1: Expansor de puertos

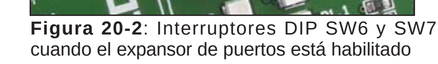


Figura 20-2: Interruptores DIP SW6 y SW7 cuando el expansor de puertos está habilitado

El microcontrolador se comunica con el expansor de puertos (circuito MCP23S17) utilizando la comunicación serial (SPI). La ventaja de esta comunicación es que sólo cuatro líneas son utilizadas para transmitir y recibir los datos simultáneamente:

MOSI - Master Output (Salida de maestro), Slave Input (Entrada de esclavo) - salida del microcontrolador, entrada de MCP23S17
MISO - Master Input (Entrada de maestro), Slave Output (Salida de esclavo) - entrada de microcontrolador, salida de MCP23S17
SCK - Serial Clock (Reloj de sincronización) - señal de reloj del microcontrolador
CS - Chip Select (Selección de chip) - habilita la transimisión de datos

La transmisión de datos se realiza en ambas direcciones simultáneamente por medio de las líneas MOSI y MISO. La línea MOSI se utiliza para transmisión de datos del microcontrolador al expansor de puertos, mientras que la línea MISO transmite los datos del expansor de puertos al microcontrolador. El microcontrolador inicializa la transmisión de datos cuando el pin CS es llevado a bajo (0V) El microcontrolador envía la señal de reloj (SCK) e inicia el intercambio de datos.

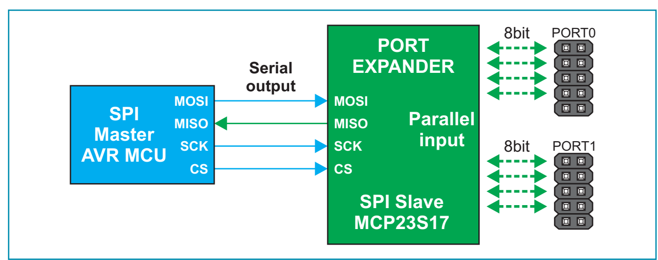


Figura 20-3: Diagrama de bloque de la comunicación SPI

El principio de funcionamiento de los puertos 0 y 1 del expansor de puertos es casi idéntico al funcionamiento de otros puertos en el sistema de desarrollo. La única diferencia yace en que las señales de los puertos se reciben en formato paralelo. El MCP23S17 covierte estas señales al formato serial y se las envía al microcontrolador. El resultado es el número reducido de las líneas utilizadas para enviarle las señales de los puertos 0 y 1 al microcontrolador.

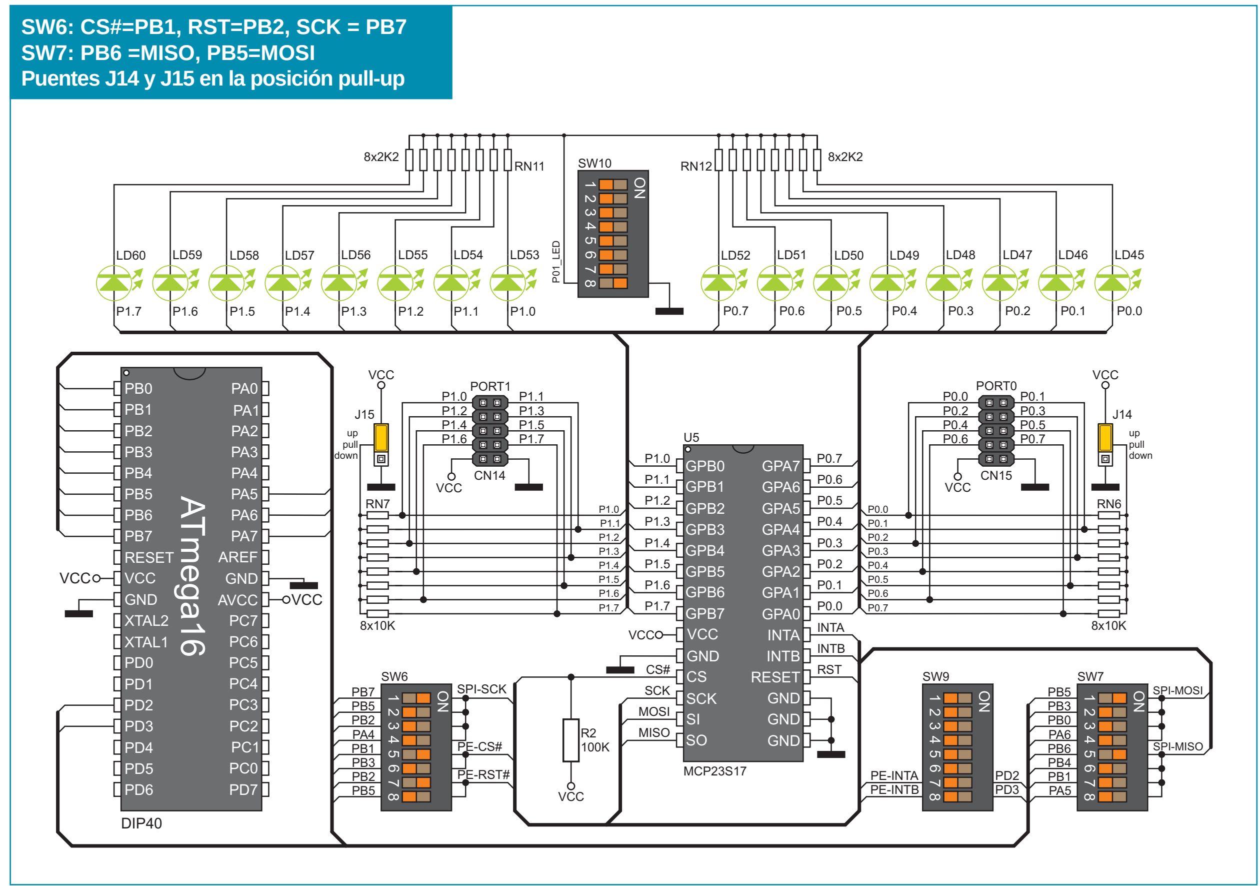


Figura 20-3: Esquema de conexión del expansor de puertos

TÉRMINOS Y CONDICIONES

MikroElektronika proporciona este manual “como está” sin garantías de ninguna especie, sean expresas o implícitas, incluyendo las garantías o condiciones implícitas de comerciabilidad y aptitud para fines específicos.

Aunque MikroElektronika ha puesto el máximo empeño en asegurar la exactitud de la información incluida en este manual, no asume la responsabilidad de ninguna especie de daños derivados del acceso a la información o de los programas y productos presentados en este manual (incluyendo daños por la pérdida de los beneficios empresariales, información comercial, interrupción de negocio o cualquier otra pérdida pecuniaria).Las informaciones contenidas en este manual son para el uso interno. Pueden ser modificadas en cualquier momento y sin aviso previo.

ACTIVIDADES DE ALTO RIESGO

Los productos de MikroElektronika no son tolerantes a fallos y no están diseñados, fabricados o pensados para su uso o reventa como equipo de control en línea en entornos peligrosos que requieran un funcionamiento sin fallos, como en instalaciones nucleares, en la navegación aérea o en sistemas de comunicaciones, de tráfico aéreo, máquinas de auxilio vital o sistemas de armamento, en los que un fallo del software podría conducir directamente a la muerte, lesiones corporales o daños físicos o medioambientales graves (“Actividades de alto riesgo”). MikroElektronika y sus proveedores niegan específicamente cualquier garantía expresa o implícita de aptitud para Actividades de alto riesgo.

MARCAS REGISTRADAS

Los productos y los nombres corporativos utilizados en este manual son protegidos por la ley de los derechos de autor, sin reparar en la ausencia de notas adicionales. Las marcas registradas son utilizadas exlusivamente con el propósito de identificar y explicar los conceptos correspondientes y en beneficio de sus respectivos propietarios, sin intención de infringirlas.

Si quiere saber más de nuestros productos, por favor visite nuestra página web www.mikroe.com

Si tiene problemas con cualquiera de nuestros productos o sólo necesita información adicional, deje un ticket en www.mikroe.com/en/support

Si tiene alguna pregunta, comentario o propuesta de negocio, póngase en contacto con nosotros en office@mikroe.com